IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Number	09/784761	**Docket Number**	BAN.0103
Filed	02/15/01	**Group Art Unit**	2154
Examiner	CHAD ZHONG	**Customer No.**	23669
Application Title	INFINIBAND WORK QUEUE TO TCP/IP TRANSLATION		
First Named Inventor	CHRISTOPHER J. PETTEY		

AFFIDAVIT UNDER 37 CFR § 1.131

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In the instant Office Action, the Examiner indicated that claims 1-45 are pending in the application and that claims 1-45 are rejected.

The Examiner rejected claims 1-6, 8-13, 15-21, 23-34, 37, 40, and 43-45 under 35 USC 102(e) as being anticipated by Beukema et al., US 2002/0073257 (hereinafter, Beukema). The Examiner also rejected claims 7, 14, 22, 35-36, 38-39, and 41-42 under 35 U.S.C. 103(a) as being unpatentable over Beukema in view of Official Notice.

Applicant respectfully disagrees with the Examiner's characterization of Beukema in consideration of both the 35 U.S.C. 102(e) and 35 U.S.C. 103(a) rejections, as detailed in the instant office action. However, rather than presenting arguments to overcome the Examiner's rejections of claims 1-45, Applicant has chosen to defer such arguments, since Applicant can demonstrate a date of invention prior to Beukema's date of invention.

More specifically, the Beukema application was filed on 12/7/2000. Applicant filed his application on 2/15/2001, less than three months later. Prior to 12/7/2000, the inventor of the present application conceived and reduced to practice his invention. At least as early as 6/30/2000, the inventor had communicated to his company his invention, and had communicated an invention disclosure to document the invention. This is evidenced by the invention disclosure transcript form prepared by the undersigned practitioner showing

the inventor's name and including a description of the invention therein. This disclosure was subsequently submitted to the parent company for approval for filing as a non-provisional patent application. Preparation for the present application was begun by the undersigned in the fourth quarter of 1999, but was not completed to be signed off on by the inventor until February of 2000.

Applicant earnestly requests the Examiner to telephone the undersigned practitioner at the telephone number provided below if the Examiner has any questions or suggestions concerning the application or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: **EO 002 739 505 US**
DATE OF DEPOSIT:**9/29/2004**
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop **AMENDMENT**, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,

HUFFMAN PATENT GROUP, LLC

By _____
RICHARD K. HUFFMAN
Reg. No. 41,082
Tel.: (719) 575-9998

Date 9/29/04

Attachment: Invention Disclosure Transcript - BAN.0103

INVENTION DISCLOSURE TRANSCRIPT - BAN.0103

Interview with Chris Pettey on 063000

The Work queue disclosure – BAN:0103

So this is a big one. So this is where we go down and we look at how we want to move data from an IB network to a mac/Ethernet/ip/tcp environment. So, there are 3 levels of connection that we need to think about. So lets just talk about a client server model and how do I connect. There are 3 levels. There's, and I am going to talk about Ethernet b/c I know Ethernet, so I have the layer 2 connection. So this mac address, here, I have 38 bits, speaketh to this mac address here 38 bits. That is how I communicate in an Ethernet environment today.

I have a layer 3 connection which says this IP address, today 32, but also available in 64, speaketh to this IP address, 32 or 64 (note: internal argument about 128, concluding on 64/128). On top of that, I get one more, and this is OS dependent as to the exact manipulation, but in TCP, there is a concept of a "connection". And this is what I open in TCP, I have a connection. There is a socket, so there are lots of ways to express a connection, you can call it a socket, whatever you want, a port, right, so you can call it connection, socket, ports, right, there are well known ports in TCP, that concept exists. So these are the 3 levels of connection that I have.

The question is, these all apply when I have a connection from that node to this node, and I have a PCI bus going. So, now when I am a network interface, where this is IB, and speaketh in IB, and this is Ethernet, and also has IP/TCP running out there, how do I associate that, and express that, with N number of servers out there, and how do I accelerate that in silicon?

So the idea is, on IB we have the concept of DLID's (sp?, I think I heard him right), doesn't work if DLID's change and we go across subnets. We have IP addresses, not too bad, but IP addresses here, are not necessarily IP addresses here. This is Ipv6, so this is Ipv whatever. So, ahhh, it doesn't really work well, and besides it only gets us to one end node.

So lets take it to the next level. We have a work queue, and a work queue (WQ) is the ultimate in connection in IB. A work queue is what the application drives. So, lets take each one of these levels, and say I want to have a hardware mapping, or I guess you could have any type of mapping, you could do this in software as well), that says, when I get a MAC address here, I can associate that with a work queue, and ship it over to this guy. So this MAC address is normally assigned to this work queue.

I can take it to the next level and say, when I get an IP address from here, that is normally assigned to this work queue.

I can take it one step further and say, I can take a connection, a port, a socket, here, and associate it with a WQ and translate it directly. So what does that get you?

So at 3 different levels, we say MAC header to WQ number. We say IP header to WQ number, and it is important to distinguish that this is not the IB IP address that you see in the global routing header. This is whatever you are expressing to the rest of the world, and they are not necessarily the same.

ME: can you take a second to distinguish those for me?

So, IP addresses, you live in the external world, you speaketh IP addresses. But, in the same way that you do masking today, so today I can go buy WinProxy. And I can take one IP address, and I can mask that to multiple IP addresses behind me, by creating a socket association. That is a well known thing, that is how you do tunneling, and a whole bunch of things, that is why IPV6 has not taken off, b/c it works very well.

ME: so I just bought a router with a firewall. I have 1 IP address assigned to the router, but it can assign 16 addresses behind it that nobody knows about.

That is exactly right, and the server routes ... You can buy a piece of software that plugs in on your win98 box and do that. That is a $45 piece of software, that function is well known.

So that same concept applies here where you have this private network that you are speaking in IP address only b/c you want to cross subnets. You are not expecting to take this IP address and map it to the rest of the universe, b/c what is in the rest of the universe is not IPV6. You are only IPV6. And, even if you were IPV6, it is not necessarily true that you are going to get the full 64,000 IP addresses for your given server. So even in there you have to have translation. So what is the best way to do translation?

Well, in this environment, I want to express an IP address. That may be a cluster, that may be more, so there are lots of ways I want to express this, so lets remove that association completely. So the IP address that is in the GRH (global routing header?) on the IB network is for one purpose only, and that is to cross from one IB subnet to another IB subnet. The IP address that exists out in the LAN environment is to get to its specific device. A specific device may be one server, it may be a cluster of servers, don't know, don't care. So, what is our association for getting there, and getting there as fast as possible. So then we say the socket/port/connection to WQ ... So we are going at each layer I can provide an extended functionality. So, at the MAC, what does that provide. It provides generalized sharing. So, I get to share this network device amongst all of these guys, as long as they can create a WQ connection, and they have a MAC address, they can run the software the same way they run today. And so they get a connection to this guy who runs as an NT driver, and to the rest of the world it just looks like we have 60-80 MAC addresses off this port, and it doesn't matter, the rest of the world can handle that.

But when we go IP, now that means that I can take a server that has multiple MAC addresses and have each IP address as a separate WQ, so there is a layer of filtering. So I can provide a layer 2 filtering, where if I have multiple MAC addresses for this server, or

I want to have 1 MAC address for all of these servers, doesn't matter, I look at the IP address, and say this IP address belongs on this WQ. I can have multiple IP addresses per server, so I can do lots of things with that functionality, so we call it layer 2 filtering, and then layer 3 functions.

What does socket/port give me? It gives me the final layer. I now go directly to the application. So I can go direct application movement. So I want to include this in the patent. The fact that now when I associate port number on this, with WQ here, when an application sets up a WQ, I can now DMA a packet directly from here, into that applications buffer. Today you cannot do that in any environment. You cannot go directly into an applications buffer. Every single TCP/IP stack in existence today does a copy. They copy the data from 1 buffer, as a NIC, to the user's buffer. And that is a performance hit. So we can instantly gain 30% of the CPU back

ME: by associating the IP address with the WQ element?

Actually, the WQ is managing the transfer. We are actually at the next level up. That can only be done at the socket or port connection. So the WQ is setup by an application trying to connect to the device. So the application now is communicating directly with you in its virtual address space. So by definition, when he posts a receive buffer, you know that receive buffer is in his user space. So when he says, when you look at the way TCP moves data, there is an open buffer, and you basically have to be able to accept data in chunks, and then the next level up handles it.

So what we are doing is allowing the TCP to exist, but rather than, get a whole bunch of data, and have to copy it and assign it to different user buffers, we can use the IB mechanisms to handle that directly.

Me: This talks about the functionality that you provide. How do you go about creating those associations?

What you have is you have a mapping table. You have a mapping table that looks works both ways. If you are coming in from the LAN side, you have 3 levels of mapping, that says, MAC address, IP address, TCP socket, so you have to be able to grep those out of the packet. And then you go into an association table that says, these N MAC addresses are associated with these N work queues. These N IP addresses are associated with these N work queues. These N sockets are associated with these N work queues.

Me: Are the WQ's static?

The WQ's can be torn down and built back up. Every time a WQ is built up, you reassign the addresses.

ME: but they are not dynamic on a WQ by WQ basis. I mean, you have 2 WQ's, you don't change one of them at a time.?

Every time I build a WQ, I associate either a MAC address, IP address, or socket with that WQ. That has no affect on either other WQ's that are built up, nor any affect on any other MAC/IP/sockets that are built up.

ME: These are specific WQ's that define the associations?

Yes, with whatever IB end node we are attempting to communicate with. We can actually say that a 3rd party transfer is now setup where we DMA'ing directly into a disk drive buffer. That is completely allowable in this. So we now have a method of 3rd party file moves from a disk to a LAN.

Me: Where does that structure live?, the N work queues?

It exists inside of this network device. It could be either implemented in silicon, which is our bubble, or implemented in a software structure, that exists in a memory around a CPU, in this network device.

Me: and a network device is a switch, or a router?

You can build it with THCA (their part), so you have a channel adapter (CA), with an IB port, PCI, E-net, LAN, CPU and memory. That structure can reside right here, today, with our current CA part, it will reside here, and the CPU and software will be cranking and doing the association thing. Our goal is to siliconize this in an integrated part here, so the CPU doesn't even have to get involved. It just sets it up and we run with it.

Me: So, you get a packet in,

It would be dma'ed into system memory, the CPU would then look at these associations, and say OK this is supposed to go to this work queue number, lets just set up a dma for this CA and go. So that is how I would implement it in software, and then we take that to the next step and implement it in silicon, so that we can bypass all of this junk directly, and go straight out.

Me: well, we should write the application to cover both software and silicon embodiments, but how does silicon establish the association.

There is still a CPU, and it is creating the association at WQ create time. Then, once the association is built, the hardware handles the direct translation of the structures.

Me: So, when an IP address hits, you have hardware that takes that IP address, or MAC address, or socket, and passes that to the correct WQ?

Correct.

Me: Let's back up a second. Tell me, you want to put this part in silicon. Tell me, what is the product?

The product is going to look like this. It will have N number of IB ports, it will have N number of Ethernet ports, and either some sort of a PCI bus for a CPU, or an integrated CPU with external memory controller. So we will still need some general purpose processing functionality to allow for this, and to allow for our customers to expand on this.

Me: so you provide Ethernet, PCI, and you are a bridge?

Correct. What this allows us to do is make a better data movement model for a LAN on the IB network. If you look at this the way the guys at IB are specifying it, they want you to do IP raw. They are saying that the right way to do Ethernet movement is you sent out the full Ethernet packet, in one IB frame, and send a command in front of that of what you are supposed to do with it. So the CPU is doing all the processing of it in that case. Then, your IB to LAN device is just sitting there, taking a packet here, sticking it blindly out there, which is fine, that is a way to do it, but performance will stink relative to our type of an operation.

Me: What is your interface for creating the work queues, if that is your block of silicon. So, if we have a PCI interface, the CPU is going to come down and, it gets a work queue create, so one of these guys, one of these servers is going to ask to create a work queue to move data out to the IB network, or the LAN network, it is going to say, I just came alive, you are my Ethernet device, I need to load my driver and run with you. So, I need to connect to a work queue. Great. I just connected a work queue. Gotcha. Now I have a connection for you I am going to associate a MAC address for you. You load my driver. Oh, you are my driver! OK, hang on just a minute. Driver, do you want to do IP association or socket association? Ah, socket association, great. Every time you get a connect request in the TCP stack, running on that host, tell me, lets create a special WQ for you, and tell me what that socket number is, that port number, so that I can do that association for you. Oh, we ran out of those, lets bump back to IP address, and we will do it globally. IF we don't find the socket we are looking for, we default down to the next level. If we don't find the IP address we are looking for, we bump down to the MAC connect level. So, we can always do it, it is just a question of how much acceleration we can provide.

Me: OK. What do you want to capture?

I want to capture anybody that is doing this association. So, we can see this association by the wire model that they use. It will be more difficult to find as we go higher up, but if they dma from Ethernet LAN, and they go directly into the WQ to user space (VA), I think we have got them.

So, here is what I am trying to do. So we go from IB to Ethernet. So, we are inserting headers on the way out, and what not. You are communicating with a whole bunch of standard devices that no absolutely nothing about IB. All they know is how TCP works. So if you take a look at the Ethernet frame, it says, you have enet here, you have IP here,

you have TCP here, and you have payload here. We are going to tear all that out, and blam, go there.

So, the reason we do all 3 is b/c you may want to have more socket connections than we have hardware support for. So, if you run out, you move down the level, and you keep moving down the level. At time 0 when you come up and connect to a device, you make sure, ok, I grab an Ethernet port. And from there, we can guarantee connectivity, and provide N level of hardware acceleration.

OK

So, if you go IB to TCP, you associate a TCP port number with a WQ, bam that is us. It doesn't matter what the hack you, if your eventual media is satellite communications, if you are doing that direct association, we got you.

Me: talk a minute about how the drafters of the spec thought you would handle the bridge.

IB spec expects that you would put the Ethernet format inside of an IB header, so you put the IB thing out there, send this to the device (bridge?). It would just strip the IB stuff off, and put the Ethernet out on the wire. And it would just keep doing that blindly. So, all of the processing responsibility would then reside in the server. So the creator of this packet, would have to create all of this entity in software, and then just move it out as an IB frame.

Me: When you have an inbound Ethernet packet here, what we said was, there is no correlation with the IB addresses

I do now, b/c I have this association with the WQ level.

Me; before your invention, you don't have that association.

Today there is no Ethernet IB association. The IB committee is proposing this as the format

Me, where this IB header here is the way you direct

What this would do is, if I am an end node, all I am doing is looking for my DLID out here. I get my DLID, I scratch this, send it out, I am done. I get a packet in, I put this IB header on, and I stick it out. And I always send it to the same location.

Me: But the end device is going to say, its mine, but what creates the association b/w this IP address and something else.

So what this does is gives you a method of sharing one Ethernet link, with multiple IB devices. So even if they keep this frame format, if we do Ethernet MAC address to IB

anything, WQ creation, even if they don't do any other processing, they just keep these, and they want to share, they will most likely have to do it this way.

Me: So you have offloaded the processing of the associations from the server, now to residing in your bridge

Yes, that is a good way to put it.